Short Read Archive Overview

National Center for Biotechnology Information (NCBI)
National Library of Medicine
Version 1.0 Draft A May 1, 2009

Table of Contents

An Introduction to the Short Read Archive

The advent of massively parallel sequencing technologies has opened an incredible new vista of research possibilities — elucidation of the human microbiome, polymorphism and mutation discovery in individual genomes, mapping of protein–DNA interactions, nucleosome positioning — the list is endless. In order to achieve these research goals using the immense power of these sequencing technologies, researchers must harness this power by being able to effectively store, access, and use the enormous volume of short read data generated in massively parallel sequencing experiments.

In response to the research community's need for a resource where the vast quantity of data generated by short read experiments can be stored, quickly accessed, and manipulated down to the individual read level, NCBI, EBI, and DDBJ under the auspices of the International Nucleotide Sequence Database Collaboration (INSDC), have developed the Short Read Archive, or SRA. The SRA is a remarkable new data storage and retrieval system that not only provides a place where researchers can archive their short read data, but enables them to quickly access known data and their associated experimental

descriptions (metadata) with pin-point accuracy from the more than 100 terabytes (100,000 gigabytes) of short read data currently available in the archive.

Using SRA, researchers will be able to access the most up-to-date short read sequence data from around the world. This is possible since the INSDC effort provides for data mirroring between the participating entities through a regular data exchange between NCBI and its partners, EBI and DDBJ, who will also use the SRA design model and code libraries as the standard in their respective short read archives.

SRA Features

A Central Data Repository with Submission Flexibility

As a data archive, the SRA preserves all the content submitted from the major sequencing technologies, and can accept submissions that originate in any of these major short read formats:

- SFF (Roche 454)
- Illumina Native
- Illumina SRF
- AB SOLiD Native
- AB SOLiD SRF

Please note that additional submission formats will be supported as they become available.

The SRA allows submissions either using an interactive web-based interface or the SRA's automated submission pipeline. The web-based submission interface is intended for occasional submissions and can be accessed following a brief initial registration (only name, email and a user-created login name required — the user sets their own password in a subsequent step). The automated pipeline is intended for sequencing centers making numerous submissions, and uses XML to describe metadata and Sequence Read Format (SRF) as a common container file format. Since the SRA uses a high-speed file transfer protocol called fasp (Aspera, Inc., Emoryville, CA), users can transfer files to and from the SRA at speeds up to 400 Mbps — many times faster than ftp. Details on obtaining the free client can be found in the SRA Submission Guidelines.

Using a common, compact design that allows for the rapid and accurate retrieval of stored data, the SRA can store all massively parallel experiment data types, including:

- Intensity Data
- Reads and Associated Quality Scores
- Trimming and other Technical Information

- Experiment Metadata
- Any secondary analyses typically performed on short read data including:
    - Alignments
    - Small-scale assemblies
    - Oligo profiles

Because of its unique design, submissions to SRA are quite flexible — a user can include all of the data elements listed above, or just a subset of them; for example, a user can submit information about their study or experiment early in the project life cycle when the details of that study or experiment are decided, and then wait to submit the experimental results until later in the project life cycle. The SRA submissions process also offers user–controlled submission modification as well as a number of Hold–Until–Published (HUP) options that include:

- Hold for a Number of Days: used for certain data release policies.
- Hold Until a Specific Date: used for the scheduled release of a publication
- Hold: used when a publishing journal has not yet been determined, or the publication date has not yet been set.

Since this flexible design also allows the SRA to accept submissions of new (still under development) short read data analyses in "blob" form (virtually no internal structure), the SRA can be a "one-stop" submission resource for small projects, projects executed by automatic pipelines, and projects submitted from newer sequencing centers that have little experience interacting with NCBI.

As massively parallel technologies develop over time, the SRA's unique design flexibility also allows for the selective movement of older, less frequently used data element(s) to less expensive storage (tape or disc), or will allow for the eventual discard of the data element(s) without having to reload any of the other data associated with these redistributed or discarded element(s).

For further information about submitting to SRA, please see the SRA Submission Guidelines.

An Integrated Search and Analysis System

As mentioned in the introduction, the SRA design model allows users to retrieve massively parallel experiment data of interest down to the individual read level quickly and with pin-point accuracy. Currently, a user can access data housed within the SRA in one of two ways —the NCBI/SRA web interface and the SRA Software Development Kit (SDK).

Accessing SRA Data using the Web Interface

Since SRA metadata is indexed in the Entrez search and retrieval system, a user can access this content directly from the NCBI home page by selecting "SRA" from the drop-down list of available databases at the top of the page, entering search terms, and then accessing read/run data from links on the response page to the SRA site. A user can also search the SRA through the coordinated use of the "Browse" and "Search" tabs on the SRA home page. The SRA web interface allows the user to:

- Access any data type stored in the SRA independently of any other data type (e.g. accessing just FASTQ without the intensity data).
- Access reads and quality in parallel
- Access integrated resources that either reference or use the particular data retrieved by the user
- Retrieve data based on ancillary information and/or sequence comparisons
- Retrieve alignments in "vertical slices" (showing underlying layered data) by reference sequence location
- Review the descriptions of studies and experiments (metadata) independently of experimental data

Accessing SRA Data using the Software Development Kit

As opposed to the web-based interface for the SRA, which allows a user to access and manipulate a limited amount of data, the SRA Software Development Kit (SDK) provide APIs (Application Programming Interfaces) that allow the user to access data housed within SRA on a larger scale:

The "Read" SDK allows the user to programmatically access signal and trace data housed within SRA, and convert it from the flexible SRA format to any of these major short read formats:

- AB SOLiD Native
- FASTQ
- SFF (Roche 454) (under development)
- Illumina Native (under development)

The "Read" SDK is designed to prevent accidental modification of the data, and is optimized to provide the user with the most efficient read possible.

The “Write” SDK, on the other hand, allows a user to both read SRA data as well as convert (write) it from the major short read formats listed below into the SRA flexible format:

- FASTQ
- AB SOLiD-SRF
- AB SOLiD-Native
- Illumina SRF
- Illumina Native (under development)
- SFF (under development)

The “Write” SDK allows users to create a local archive using their own short read data, and in future will used for direct submissions to the SRA.

SRA Architecture

The Need for a New Paradigm in Massive Data Storage and Retrieval

NCBI began development of the SRA database in October, 2007 in response to the research community’s need for efficient and flexible ways to store and retrieve the large amounts of massively parallel sequencing data beginning to appear. Now that the SRA has reached an initial state of completion and is publically available at NCBI, it is currently being deployed at EBI, and will be deployed at DDBJ in the future so as to foster an efficient exchange of SRA data between these INSDC members. NCBI and EBI have already begun exchanging data, and once the SRA is in place at DDBJ, there will be a regular data exchange between all three entities.

The initial design of the SRA was conceived by looking at the advantages and disadvantages inherent in relational databases and in file-based storage systems, and using the best aspects of each to create something entirely new.

Relational Databases Alone are Not the Answer

Relational databases are good for recording and manipulating related data: they can index, make complex arbitrary joins, and they can process complex queries. Relational databases, however, are not a practical approach for the long term storage of the terabyte and petabyte amounts of data generated from massively parallel sequencing experiments because of disadvantages inherent in relational databases:

- They are inflexible
- They are slightly wasteful
- They intentionally avoid using the file system
- They encapsulate all data behind their servers

In summary, relational databases are bulky, require significant management and are inflexible and costly both in terms of licensing and use of storage space.

File-based Storage Systems Alone are Not the Answer

Although simple file-based storage systems have many advantages to recommend them for storing massive amounts of data:

- They are lightweight
- They make use of the file and directory systems already available
- They store data according to an object model (i.e. a run is in a single file, which is an object that can be accessed, shipped, modified, removed, etc.)

file-based storage systems suffer from a lack of support for indexed queries, and do not have the ability to create relationships when and where they are necessary. Archiving data in a file-based storage system often means using the tar (tape archive) format and applying a compression utility like gzip or bzip2 to produce a compressed tar file, making the data difficult (or impossible) to access in a repository setting.

A Hybrid Storage and Retrieval System

In order to store and retrieve the enormous amount of data generated by massively parallel sequencing technologies, NCBI, EBI and DDBJ needed to create a data repository that has much of the power of a relational database while being lightweight, transportable and flexible like flat-file storage. We found our solution by creating a file-based, column-oriented design – relational database hybrid.

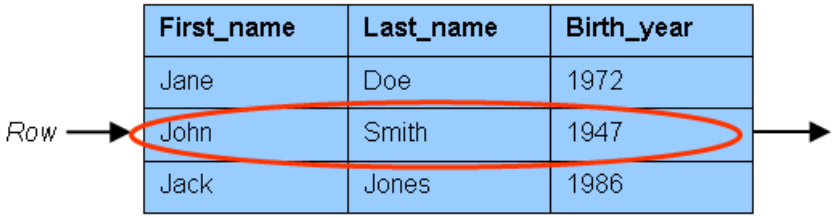
SRA: A Dynamic Mix of Database and Cloud Computing

By offering a resource that will allow a user to compute with their feet on the ground and their head in the clouds, the SRA can offer great versatility in massive data storage and management.

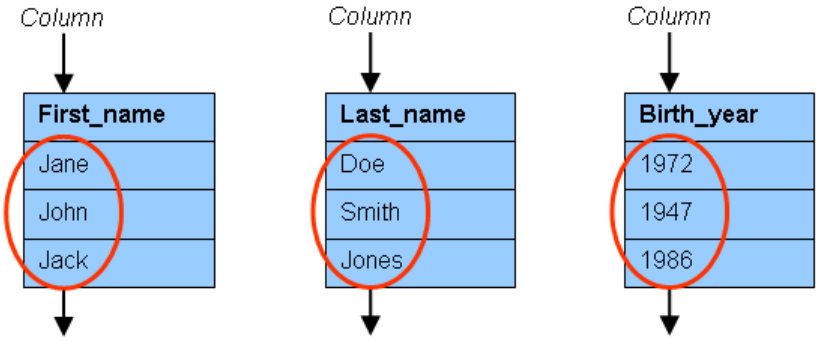
SRA is not just a format, but also a novel combination of a file-based, column-oriented design and a relational database, which allows the SRA to provide search and retrieval services that can be based not only at NCBI but also locally once a user moves their data into the SRA flexible format.

Row vs. column-oriented Database Design

Row-oriented databases store data as a series of row structures, where each structure contains one or more fields (unique data types arranged in columns) linked together in a table. In the row oriented system, a user approaches the data from left to right in a single row:



Column-oriented databases turn this structure on its side, so to speak, and store the data as columns where each data type is stored as a series independently in its own column (still associated with its unique ID). In the column-oriented design, a user approaches each data type as an independent series moving from the top of the column downward to the bottom of the column:



The advantage of row-oriented databases is that they link together all the fields (data types) in a single row so that the entire row can be retrieved with a single read. This advantage becomes a disadvantage, however, when this design is applied to the problem of massive data storage:

- Since multiple data types are involved, compression and/or packing of data in row-oriented tables is difficult and not efficient in the use of storage space.

- Addition or removal of specific fields (data type columns) is extremely difficult: since the fields are linked together in each row, the removal of one field necessitates the re-write of the entire data table

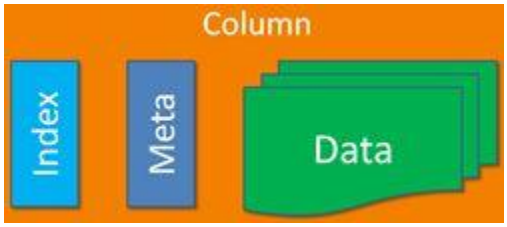
Column-oriented databases, however, take advantage of the fact that there is a single data type per column unit to achieve improved storage (with only one data type, packing and/or compression is easier and more efficient in the use of storage space) and retrieval efficiency. In addition, if a column-oriented database is properly designed, a column (data type) can be added or removed independently of the other columns in the table, leaving the remaining data intact, so there is no need to re-write an entire data table every time you want to add or remove a data type.

The File-based, Column-Oriented Design – Relational Database Hybrid: Putting your Head in the Clouds and Your Feet on the Ground

SRA's file-based, column-oriented design makes use of the file system to keep the data columns physically separate. Each data column within the SRA design model is packaged in its own file rather than in a database. This makes it possible to store the most frequently accessed data series (e.g. "FASTQ" data) in fast, near storage, and less frequently accessed data like intensities and reads in slower, bulk storage like tape (intensities) or disk (reads), located at the repository where the data were submitted (NCBI, EBI, or DDBJ). If the data is located in bulk storage at NCBI, EBI, or DDBJ, the data column/file can be transferred to the user via the compute cloud (internet) when called upon independently of other data types. This allows the SRA to be used to access and read any short read data it houses in random order, stream it quickly, or get reads and quality in parallel. The NCBI relational database portion of the SRA hybrid design model serves to track runs and components, while the SRA toolkit operates using directories and files, so it can easily bring the power of the SRA approach to a local computer.

SRA Data Structure

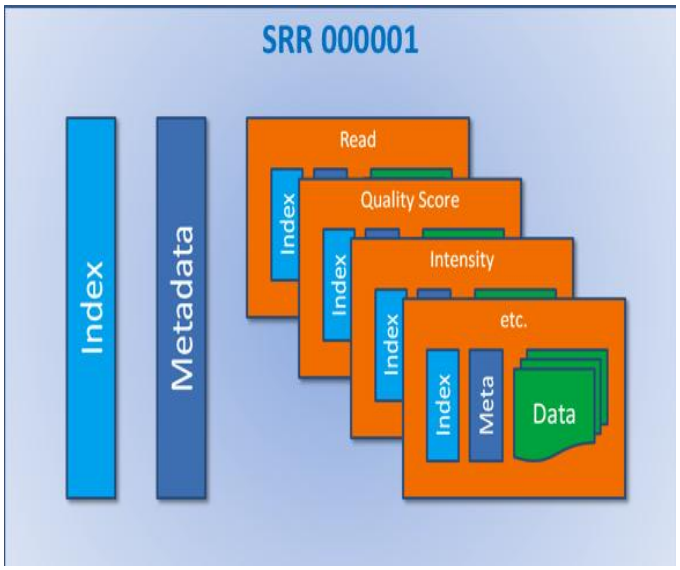
As mentioned above, SRA uses file-based data management, where the base unit is a column, packaged within its own file. Each column represents a single data type (i.e. read, quality score, or intensity data, etc.), and the file holding the column contains not only the particular data type, but also the index of identifiers for, and a minimal description (metadata) of, each member of that data type:



If for example, the column represented above holds read data for a single run within an experiment, then this "read column" (file) would contain the series of template reads (data) generated in the run, the identifiers for each read (which do not have explicit IDs, but are given serial numbers based on the run ID to save space) and a small amount of metadata that describes the read (name [alias] and plate location).

The "columns" (files) are then organized together into a "run" — a "table" in database parlance. The "run" (table) groups columns that contain the data gathered for a sample or sample bundle in a particular experiment into a single

structure (the number of columns in a run is arbitrary — it depends on the number of data types available):



In the example above, the run accession number is SRR000001; this particular run contains a series of Read data, Quality Score data, Intensity data, as well as other data types that may exist ("etc."), where each data type is contained within its own column (file). In addition, the run "table" contains a more substantial amount of metadata, including technical information about the instrument model, date of run, run center, plate statistics, brief experimental description, etc., as well as an overarching index for the data housed within the run.

A unique feature of this type of table is that it takes its runtime structural definition from the contents of the file system — that is, it determines its component columns dynamically when it is opened. This feature, therefore, provides the user with incredible versatility when it comes to archiving data since old data types can easily be removed and replaced with new data types — for example, if new quality data becomes available and replaces the quality data currently in use, the old quality data column can be removed and replaced with a new quality column, and even though the new quality data is of an entirely different type than the old column, the table will resolve — based on the new data type present. Or perhaps a user no longer wishes to archive intensity data — no problem — the intensity column can easily be removed in the same way without any replacement at all, and the table will still resolve upon opening.

The Necessity of Archival Data Storage

There is a great deal of discussion in the massively parallel sequencing community regarding the necessity of archiving the various data types generated within short read sequencing experiments:

Intensity Data

Some within the community feel that intensities are no longer needed once base calls are made. The SRA, however, is capable of archiving intensity data, so that should bases need to be re-called — for example, if new and improved base calling algorithms are developed — it will be possible to do so. Many project leaders are choosing to archive intensities for early experiments completed prior to the establishment of optimal base calling, or for important projects where it may be difficult to re-sequence the samples. It may be sufficient, however to deposit only the reads for projects such as ChipSeq experiments. For those projects depositing intensity data, the SRA provides the option to discard the intensity data at a later date when it the community deems that it is no longer necessary.

Read Data

The SRA can also archive read data since:

- Archived reads can be re-used for alignments when improved alignment algorithms are available
- Archived reads can be used to regenerate an alignment when a reference assembly is updated
- Archived reads can be used in generating an alignment to another reference assembly (e.g. European, Asian, or African reference human genome assemblies)
- Archived reads can be used to pool data across different experiments or create experimental sub-sets from within an experiment

Alignment Data

Finally, the SRA intends to archive alignment data since:

- Archived Alignments can be re-analyzed for purposes different from those intended in the original experiment (e.g. alignment data from a gene expression experiment could be used for SNP verification)
- Archived Alignments can be used to verify simple summaries like histograms.
- Archived Alignments can be used to generate SNP calls, CNV calls, or different histograms

SRA Future Developments

The SRA will continue to support new platforms and sequencing technologies as they become available, and is currently developing an archival storage system for short read alignment and assembly records.